REMARKS

Claims 1 and 4-21 are now pending. Reconsideration is respectfully requested.

35 U.S.C. § 103 Rejections

Claims 1, 4-5, 8-13, 15-18, and 20-21 stand rejected under 35 U.S.C. § 103(a) as unpatentable over Wentker et al. (U.S. Patent No. 6,481,632) and further in view of Muttik et al. (U.S. Patent No. 6,907,396). Applicant respectfully traverses this rejection.

The Wentker-Muttik combination does not teach or suggest each and every element of the claims. The examiner asserts that a pre-synchronization scan is equivalent to Wentker's discussion that testing is performed on an application before it is given to an issuer. Applicant respectfully submits that the discussion of testing is inconsistent with the broadest reasonable construction consistent with the specification.

Only through unreasonable construction of the claims/specification is Wentker relevant

While the claims are to be given the broadest reasonable construction, the broadest reasonable construction is one "read in the appropriate context of the claim language and specification." *In re Suitco Surface, Inc.*, 2009-1418 (Fed. Cir. 2010) (stating "[t]he broadest-construction rubric coupled with the term "comprising" does not give the PTO an unfettered license to interpret claims to embrace anything remotely related to the claimed invention. Rather, claims should always be read in light of the specification and teachings in the underlying patent.").

In the present case, a pre-synchronization scan is performed after loading the software on an open platform computer system. In response to Applicant's arguments, the examiner asserts

> A synchronization phase is merely a phase where the software in one device may be transferred to another so that both have the same information. Therefore, pre-synchronization (as used in the scope of the claim) would be any phase before the synchronization/loading phase occurs. Furthermore, Applicants only specify that the software is loaded on the open platform system and does not specify which element of the open platform system comprising of a host facility and portable computing device (as well as other possible nodes within the system) actually loads the software in a manner that initiates a pre-synchronization scan. Final Office Action, 2/4/11, p. 2.

The examiner is unreasonably construing an open platform computer system as any set of activities possibly connected to the system, such as a provider testing its software. The limitation of an open platform computer system was included in Applicant's November 13, 2008 response to office action. The amendment was to distinguish the current invention as claimed from platforms that are closed platform computer systems (e.g., Mohammed and Brody, already of record). As discussed in the November 13, 2008 response, closed platforms are those platforms whereby the system is only open to those possessing the required credentials to develop for the system. In these cases, a network service provider or a software publisher retained complete control of the system and its security features.

Wentker's system is closed because it discloses taking ownership of security domain, the use of secret keys, and assigning secret keys to a provider. If a developer does not belong to this system, their application does not make it to a smart card. Wentker has the benefit of always being able to rely on an application provider being a trusted third party. Wentker discloses

> Later in the life of a card when a new application provider wishes to take ownership of a security domain on the card and use it to load an application, this provider receives a secret key set for one of the unassigned security domains from the trusted third party. In this way, the key set for a security domain is known only to the new application provider and is kept separate from the issuer or another application provider.
>
> In another example of how a security domain and its secret keys may be assigned to a provider, a technique as described in U.S. patent application Ser. No. 09/046,993 may be used. In this scenario, one of the security domains on the card is assigned to a trusted third party. The trusted third party takes ownership of the domain and control of its secret keys. Once the card is issued and is in use by a cardholder, the trusted third party retains ownership of the security domain. A new application provider that wishes to load an application onto the card would then approach the trusted third party for permission to use their security domain. The application provider would then load their own application (which might be a new security domain) using the security domain and secret keys of the trusted third party. In this fashion, a new application provider is allowed to load a new application and security keys without having to share the same with an issuer or another application provider. These techniques and others may be used to provide a security domain on a smart card having a security key set that is known to the application provider to whom the security domain belongs. Col. 13, lines 32-60.

An open platform computer system has the meaning that a vendor or distributor or similar entity does not retain control of the security features of the computer system. This meaning is

consistent with the specification and the claimed invention, which is fulfilling the need of a "computer manufacturer to operate under an open platform system yet prevent the inadvertent installation of Trojan horses (and viruses) as part of the software used in a palmtop computing device." Specification, p. 5-6. In other words, the manufacturer of a portable computing device does not retain control of the security features of the device after it is purchased. This meaning cannot simply be dismissed nor can it be construed to include any possible node connected to the open platform system. Wentker is clearly discussed in the context of a closed platform system similar to Mohammed and Brody already of record. The issuer clearly controls the platform through its delegated loading functionality.

When read within the proper context, synchronization as used by the Applicants' is not merely transferring software from one device to another. The examiner asserts an unsupported definition of the term synchronization. Synchronization as used within the specification and the claims means the well-known synchronization processes that occur between a host facility and a portable computing device. Therefore, the present claims must be construed under the context of a synchronization process between an open platform computer system comprises a host facility and a portable computer device.

Wentker does not use the term synchronization anywhere within its specification nor does Wentker attempt to describe a process of synchronization between an open platform computer system comprising a host facility and a portable computing device. Wentker's invention is specifically applicable to smart cards. Wentker's invention enables an issuer of a smart card, for example, a credit card company, to provide limited management capabilities (i.e., delegated management) to application providers to load, install, and delete their application on the smart card. These changes are pre-approved by the smart card issuer to increase the flexibility in managing the smart card. In this system, the application provider may prevent the issuer from accessing private user data on the card because they are delegated separate security domains to manage their application. The issuer can simply pre-approve certain applications by providing a command authentication pattern to the application provider. Upon loading an application on a smart card, the command authentication pattern is verified. See e.g., col. 2, line 65 – col. 3, line 40. Loading software on a smart card is not a synchronization process. The delegated loading process is merely loading an approved application onto the card. This process contains little or

none of the attributes attributable to well-known synchronization processes. Wentker's process differs significantly from the present claims.

An application provider testing software is not equivalent to a pre-synchronization scan

A pre-synchronization scan is a scan that occurs before the synchronization process yet after the software has been loaded on the open platform system comprising a host facility that directly will be synchronizing with the portable computing device. The present invention as claimed utilizes a validator program that scans and validates software by running the software in an emulator in a secure environment. The secure environment comprises a modified operating system for the emulator to run in so that the code may be examined for malicious routines such as viruses, Trojans and the like.

The examiner's construction is not a reasonable construction. The examiner states

> Applicants only specify that the software is loaded on the open platform system and does not specify which element of the open platform system comprising of a host facility and portable computing device (as well as possible other nodes within the system) actually loads the software in a manner that initiates a pre-synchronization scan. Final Office Action, 2/4/2011, p. 2.

A reasonable construction consistent with the specification and claims is that the software to operate on the open platform computer system is loaded on the host facility. The host facility is the host associated with the portable computing device because should the software be found invalid, it is this host facility that will deny synchronization with the portable computing device.

The examiner alleges that the activities of an application provider are equivalent to pre-synchronization scans. The examiner cites in substance

> Testing of an application for a smart card may be performed in any of a variety of ways and is a step understood in the art, and generally involves functional tests (optional) and security tests (mandatory). Testing of the application involves checking its operational behavior on a smart card, checking its operational memory requirements, etc., ensuring that the application is secure, and checking for viruses and card related threats. Once the issuer (or trusted third party) has tested the application and it to ensure that it behaves correctly, the application is "certified" and the issuer is ready to prepare the application for a delegated load and installation by the provider. Col. 15, lines 8 – 19.

These activities are all performed prier to the delegated load and installation conducted by the provider not by a validator program residing in the open platform computer system. All these

activities occur outside the open platform computer system because the system disclosed is that of a closed system. The examiner appears to skip over the existence of a validator program existing on the open platform computer system by stating

> It is clear from the previous citation that Wentker et al. teach utilization of some type of validation program running within the system (again, within any element of the open platform system since the claims fail to specify) in order to check for potentially harmful behavior within each application. Final Office Action, 2/4/11, p. 3.

The examiner cannot point to a validation program and must resort to declaring it "some type of validation program running within the system." This amounts to a mere guess of what the issuer's or provider's activities actually are. Nowhere is running an emulator mentioned.

Neither citation references a pre-synchronization scan, a validator program, an emulator, or scanning the software by the validator program in a secure environment. With respect to these comments regarding the "pre-synchronization" activities asserted by the examiner, Wentker's disclosure has absolutely no relevance to the claims at hand. These citations are within a section describing Fig. 7A, which is a flow diagram describing a technique for delegated loading. Wentker describes delegated loading as "allow[ing] the application provider to establish a loading session for transferring their application files directly to their own security domains." Col. 12, lines 29 – 31. The data authentication pattern is merely a mechanism used to ensure the authenticity of the software. In other words, to ensure that the software approved by the issuer is the same software being loaded.

Wentker's delegated loading process is not equivalent to a pre-synchronization scan

The examiner states that "upon loading the software on the open platform computer system, initiating a pre-synchronization scan" and the acts of scanning and validating is equivalent to:

> The card issuer pre-authorizes the initial install command (which performs loading) and the load file through the use of these data authentication patterns. The data authentication pattern for the application file is included in the initial Install command to ensure that application which has been approved by the card issuer is the same application that is subsequently received by the card manager through the series of loading commands that follow the first install command. Col. 12, lines 49 – 57.

Applicant disagrees. By relying on proper constructions of the terms an open platform computer system and pre-synchronization scan, the cited disclosure is not equivalent to the assert claim limitation. As stated above, Figure 7A discloses that the install commands ensures the application is the same application that is received by the card manager. If the application is the same, the application is made available through the delegated loading process. This is not a scan for malicious routines.

Notwithstanding the fact that Wentker contains no concept of a pre-synchronization scan, at no time is the software to be loaded scanned or validated during a pre-synchronization scan by an emulator. The testing referred to by the examiner is outside the context and scope of the claims. Wentker comtemplates that an issuer may have a third party test an application on a smart card to make sure it runs properly. This testing step is divorced from any process related to Wentker's invention, but is merely an acknowledgement by the inventor that the software industry generally tests its software before distribution. Even if this testing could be construed as being relevant disclosure as the examiner asserts, there is no mention of running an emulator in a modified operating system so that the code may be examined for malicious routines as claimed.

Furthermore, the above passage does not state that the software is marked with a flag during any validation process within the proper context and scope of the claims that would possibly deny the software the ability to run on the system and deny synchronization. Again, notwithstanding that Wentker does not disclose a synchronization process, if the software would be tested by a third party and would not meet the issuer's specifications, it is likely that the software would be fixed by the application provider so that the software would run. There would be no conceivable reason for a third party to test a piece of software only to mark the software as unusable and still enable it to be distributed to the issuer for use on a smart card. This is the benefit of a closed system.

Given the context of Wentker's disclosure, this scenario tests the limits of reasonable construction as outlined above. The examiner states

> Examiner would like to point out that the system of Wentker et al. comprises several phases when an application is developed, where the first phase is testing the software, another step is authenticating the source/integrity, and a final state is synchronization. Therefore, testing a piece of software within the open platform system disclosed in Wentker et al. allows for a step of ensuring that the application behaves appropriately, and if it does marking it as "certified" (col. 15,

> lines 15-19). On the contrary, if it is determined that it is not behaving as it should, it is denied from entering future phases until it has been fixed to perform properly (col. 9, lines 63-65). Therefore, it is not the case where the software is tested, marked as unusable, and still distributed in the unusable form to future stages of the process disclosed in Wentker et al. Examiner would also like to note that the claimed limitations do not limit Examiner from interpreting the pre-synchronization phase as the application testing phase since it occurs before synchronization within the open platform system. Furthermore, the claims do not limit the scope of the invention to a state where another validation check/scan is implemented after the software industry has already done their check and directly before loading the software onto the host. Again Examiner would like to emphasize that the "pre-synchronization scan" as claimed does not state an exact order of where the pre-synchronization scan occurs within the system, it only indicates that it occurs before the synchronization phase. Final Office Action, 2/4/11, p. 4.

The examiner has unreasonably construed the activities of an entire software industry to be the pre-synchronization scan that is initiated after loading the software on the open platform computer system comprising a host facility and a portable computing device. The issue presented by the present invention as claimed is that the entire software industry and their activities cannot be trusted unless developed under a closed system such as Wentker, Mohammed and Brody, already of record. The development of the closed system limits the availability of third party software for a computer manufacturer's device. Wentker's process is inapplicable to a system where third party developers are neither known nor controlled by the manufacturer. The present invention as claimed solves this issue.

When the claims and specification are reasonably construed within their proper contexts, the claims do limit the scope of the invention to a state where another validation check/scan is implemented after the software industry conducts its activities. It is unclear how the limitation "upon loading the software on the open platform computer system, initiating a pre-synchronization scan" does not state where and when the scan occurs. It certainly does not explicitly state or imply that the scan occurs on a developer's machine possibly half way around the world from the loading of the software on a user's system. Such broad constructions are not consistent when read against the instant specification.

Muttik's combination is improper due to impermissible hindsight

The suggested combination of Wentker with Muttik is improper. Wentker provides absolutely no motivation, teaching, or suggestion for one ordinarily skilled in the art to consult Muttik (or any reference discussing emulator's for that matter). Muttik teaches improving an already existing emulator by patching additional instructions (i.e., extensions) into the emulator. See e.g., Abstract, col. 1, lines 46 – 50, and col. 2, lines 10 – 15. Muttik does not discuss situations where an emulator would be advantageous or disadvantageous as emulators existing before Muttik. Muttik's invention assumes that an emulator on a system exists and that using extensions is an improvement to that security technique. So any suggestion that emulation would be a valuable technique to combine with Wentker comes solely, and impermissibly, from the Applicant's disclosure.

To counter this argument, the examiner states

> In this case, Muttik et al. specifically state "Emulator buffer 201 and emulator code 203 are designed so that while suspect code 108 that is executing within emulator buffer 201, suspect code 108 cannot damage or compromise computer system 106" (col. 4, lines 18-22). Final Office Action, 2/4/11, p. 6.

This passage merely suggests a definition of emulation. There is no citation in Wentker or Muttik that points to a suggestion that the emulation technique would motivate one to combine it with Wentker's process. Muttik's disclosure is about making existing emulator's better. Wentker does not disclose or utilize an emulator. Wentker leaves any technique for validating code much less running one in an emulator, which is not mentioned by Wentker, as an exercise for developers outside of the delegated loading process. Stating the definition or function of an emulator says nothing about its combinability with other references. The motivation to combine emulation in the manner claimed comes solely from the Applicant's disclosure. Therefore, combination is impermissible.

Furthermore, Muttik's scope does not include and thus does not disclose "a computer system comprising a host facility and a portable computer device coupled to the host facility" (emphasis added). In other words, like Wentker, Muttik has no concept of a synchronization process and the challenges faced by the use of such a process. Muttik is directed to a single system performing its functionality. See, e.g., col. 3, lines 43-49. Muttik does not disclose a system wherein a host and a portable device are operating in concert to achieve the claimed

functionality. It is irrelevant that Muttik was not relied upon to teach this functionality. The fact that Muttik does not disclose this feature evidences that Muttik contains no suggestion to combine with Wentker and vice versa. Therefore, any combination with Wentker must come from the Applicant's disclosure.

As argued above, one ordinarily skilled in the art would not look to combine these references because of their quite disparate teachings. Applicant's claims and specification are being given an unreasonable construction if they are read in their proper context. The unreasonable construction leads to additional unreasonable equivalencies between the functions disclosed in Wentker and elements of the claims. Wentker does not mention the use of emulators, yet somehow Muttik, which is directed towards improving existing emulators, has been suggested to close this deficiency. To combine Muttik with Wentker, the disclosed smart card process would need to be modified to include an emulator. Neither reference discloses a pre-synchronization or synchronization process so neither reference is capable of disclosing denying a synchronization of software. Therefore, Wentker and Muttik, alone or in combination, do not disclose, teach, or suggest each and every element of the claims as required. Accordingly, Applicant respectfully requests withdrawal of this rejection.

Claim 7 stands rejected under 35 U.S.C. § 103(a) as unpatentable over Wentker and Muttik as applied to claim 1, and further in view of Brody et al. (U.S. Pub. No. 2001/0051928). Applicant respectfully traverses this rejection.

As argued with regard to claims 1, Wentker and Muttik, alone or in combination, do not teach or suggest the present claims. Brody does not cure the Wentker-Muttik combination's deficiencies. As argued in previous responses, it is unclear how Brody has any relevance to Muttik and now it is equally unclear how Brody has any relevance to Wentker. Brody adds nothing to the combination and as such the combination still does not teach or suggest the claim. Any motivation to combine these references comes solely from the Applicant's disclosure. Accordingly, Applicant respectfully requests withdrawal of this rejection.

Claims 6, 14, and 19 stand rejected under 35 U.S.C. § 103(a) as unpatentable over Wentker and Muttik as applied to claims 1, 8, & 18, further in view of Ginter et al. (U.S. Patent No. 6,948,070). Applicant respectfully traverses this rejection.

As argued with regard to claims 1, 8, and 18, Wentker and Muttik, alone or in combination, do not teach or suggest the present claims. Ginter does not cure the Wentker-Muttik combination's deficiencies. Ginter is directed towards electronic commerce transactions and has little or no relevance to either Wentker or Muttik. Any motivation to combine these references comes solely from the Applicant's disclosure. Accordingly, Applicant respectfully requests withdrawal of this rejection.

Conclusion

In light of the above remarks, Applicant respectfully requests reconsideration of the rejected claims and solicits their allowance. In the event an interview is useful in resolving any issues, the examiner is invited to telephone the undersigned representative.

Respectfully submitted,

BERRY & ASSOCIATES P.C.

Dated: August 2, 2011

By: /Shawn Diedtrich/
Shawn Diedtrich
Registration No. 58,176
Direct: 480.704.4615

9229 Sunset Blvd., Suite 630
Los Angeles, CA 90069
(310) 247-2860